ANNUAL REVIEWS

Annual Review of Statistics and Its Application

A Survey of Tuning Parameter Selection for High-Dimensional Regression

Yunan Wu and Lan Wang

School of Statistics, University of Minnesota, Minneapolis, Minnesota 55455, USA;
email: wangx346@umn.edu

Annu. Rev. Stat. Appl. 2020. 7:209–26

The *Annual Review of Statistics and Its Application* is online at statistics.annualreviews.org

https://doi.org/10.1146/annurev-statistics-030718-105038

Keywords

tuning parameter, lasso, cross-validation, scaled lasso, square-root lasso, BIC, Bayesian information criterion

Abstract

Penalized (or regularized) regression, as represented by lasso and its variants, has become a standard technique for analyzing high-dimensional data when the number of variables substantially exceeds the sample size. The performance of penalized regression relies crucially on the choice of the tuning parameter, which determines the amount of regularization and hence the sparsity level of the fitted model. The optimal choice of tuning parameter depends on both the structure of the design matrix and the unknown random error distribution (variance, tail behavior, etc.). This article reviews the current literature of tuning parameter selection for high-dimensional regression from both the theoretical and practical perspectives. We discuss various strategies that choose the tuning parameter to achieve prediction accuracy or support recovery. We also review several recently proposed methods for tuning-free high-dimensional regression.

1. INTRODUCTION

High-dimensional data, where the number of covariates/features (e.g., genes) may be of the same order as or substantially exceed the sample size (e.g., number of patients), have become common in many fields due to advancements in science and technology. Statistical methods for analyzing high-dimensional data have been the focus of an enormous amount of research in the past decade or so; readers are directed to the books of Hastie et al. (2009), Bühlmann & Van de Geer (2011), Hastie et al. (2015), and Wainwright (2019), among others, for extensive discussions.

In this article, we consider a linear regression model of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0 + \boldsymbol{\epsilon}, \qquad 1.$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the vector of responses, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ is an $n \times p$ matrix of covariates, $\boldsymbol{\beta}_0 = (\beta_{01}, \ldots, \beta_{0p})^T$ is the vector of unknown regression coefficients, and $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T$ is a random noise vector with each entry having mean zero and variance σ^2. We are interested in the problem of estimating $\boldsymbol{\beta}_0$ when $p \gg n$. The parameter $\boldsymbol{\beta}_0$ is usually not identifiable in high dimension without imposing additional structural assumptions, as there may exist $\boldsymbol{\beta}_0' \neq \boldsymbol{\beta}_0$ but $\mathbf{X}\boldsymbol{\beta}_0' = \mathbf{X}\boldsymbol{\beta}_0$. One intuitive and popular structural assumption underlying a large body of the past work on high-dimensional regression is the assumption of strong (or hard) sparsity. Loosely speaking, it means only a relatively small number—usually much less than the sample size n—of the p covariates are active in the regression model.

To overcome the issue of overfitting, high-dimensional data analysis employs penalized or regularized regression techniques represented by lasso (Tibshirani 1996, Chen et al. 2001) and its variants such as the Dantzig selector (Candes & Tao 2007), smoothly clipped absolute deviation (SCAD) (Fan & Li 2001), minimax concave penalty (MCP) (C.H. Zhang 2010) and capped L_1 (T. Zhang 2010). In a nutshell, a high-dimensional penalized regression estimator solves

$$\min_{\boldsymbol{\beta}} \left\{ (2n)^{-1} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \sum_{j=1}^{p} p_\lambda(|\beta_j|) \right\}, \qquad 2.$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$, $||\cdot||$ denotes the L_2 vector norm, and $p_\lambda(\cdot)$ is a penalty function that depends on a tuning parameter $\lambda > 0$. Customarily, the intercept β_0 is not penalized.

Regardless of the penalty function, the choice of the tuning parameter λ plays a crucial role in the performance of the penalized high-dimensional regression estimator. The tuning parameter λ determines the level of the sparsity of the solution. Generally speaking, a larger value of λ indicates a heavier penalty and tends to produce a sparser model.

This article aims to provide a broad review of the current literature on tuning parameter selection for high-dimensional penalized regression from both theoretical and practical perspectives. We discuss different strategies for tuning parameter selection to achieve accurate prediction performance or to identify active variables in the model; the latter goal is often referred to as support recovery. We also review several recently proposed tuning-free high-dimensional regression procedures, which circumvent the difficulty of tuning parameter selection.

2. TUNING PARAMETER SELECTION FOR LASSO

2.1. Background

A simple yet successful approach for avoiding overfitting and enforcing sparsity is to regularize the classical least-squares regression with the L_1 penalty, corresponding to adopting $p_\lambda(|\beta_j|) = \lambda|\beta_j|$ in

Expression 2. This choice leads to the well-known least absolute shrinkage and selection operator (lasso) (Tibshirani 1996), which simultaneously performs estimation and variable selection. In the field of signal processing, the lasso is also known as basis pursuit (Chen et al. 2001).

Formally, the lasso estimator $\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda)$ is obtained by minimizing the regularized least-squares loss function, that is,

$$\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \left\{ (2n)^{-1} \sum_{i=1}^{n} (Y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda ||\boldsymbol{\beta}||_1 \right\}, \quad 3.$$

where $\mathbf{x}_i^T = (x_{i1}, \ldots, x_{ip})$ is the ith row of $\mathbf{X}$, $||\boldsymbol{\beta}||_1$ denotes the L_1-norm of $\boldsymbol{\beta}$ and λ denotes the tuning parameter. By varying the value of λ and solving the above minimization problem for each λ, we obtain a solution path for lasso.

In the literature, a great deal of work has been devoted to understanding the theoretical properties of lasso, including the theoretical guarantee on the nonasymptotic estimation error bound $||\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda) - \boldsymbol{\beta}_0||_2$, the prediction error bound $||\mathbf{X}(\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda) - \boldsymbol{\beta}_0)||_2$, and the ability to recover the support set or the active set of the model $\{j : \beta_{0j} \neq 0, j = 1, \ldots, p\}$ (see Greenshtein & Ritov 2004, Meinshausen & Bühlmann 2006, Zhao & Yu 2006, Bunea et al. 2007, Van de Geer 2008, Zhang & Huang 2008, Bickel et al. 2009, Candès & Plan 2009, among others). The tremendous success of L_1-regularized regression technique is partly due to its computational convenience. Efficient algorithms such as the exact path-following least angle regression (LARS) algorithm (Efron et al. 2004) and the fast coordinate descent algorithm (Friedman et al. 2007, Wu & Lange 2008) have greatly facilitated the use of lasso.

2.2. A Theoretical Perspective for Tuning Parameter Selection

Motivated by the Karush–Kuhn–Tucker condition for convex optimization (Boyd & Vandenberghe 2004), Bickel et al. (2009) proposed a general principal for selecting λ for lasso. More specifically, they suggested that λ should be chosen such that

$$\mathrm{P}\left\{ ||n^{-1}\mathbf{X}^T\boldsymbol{\epsilon}||_\infty \leq \lambda \right\} \geq 1 - \alpha, \quad 4.$$

for some small $\alpha > 0$, where $||\cdot||_\infty$ denotes the infinity (or supremum) norm.

Consider the important example where the random errors ϵ_i, $i = 1, \ldots, n$, are independent $N(0, \sigma^2)$ random variables and the design matrix is normalized such that each column has L_2-norm equal to $\sqrt{n}$. One can show that an upper bound of λ satisfying Equation 4 is given by $\tau\sigma\sqrt{\log p/n}$ for some positive constant τ. To see this, we observe that by the property of the tail probability of Gaussian distribution and the union bound,

$$\mathrm{P}\left\{ ||n^{-1}\mathbf{X}^T\boldsymbol{\epsilon}||_\infty \leq \tau\sigma\sqrt{\log p/n} \right\} \geq 1 - 2\exp\left(-(\tau^2 - 2)\log p/2\right),$$

for some $\tau > \sqrt{2}$. Similar probability bound holds if the random errors have sub-Gaussian distributions (e.g., section 4.2 of Negahban et al. 2012).

Most of the existing theoretical properties of lasso were derived while fixing λ at an oracle value satisfying Equation 4 or within a range of oracle values whose bounds satisfy similar constraints. For example, the near-oracle error bound of lasso given by Bickel et al. (2009) was derived assuming $\lambda = \tau\sigma\sqrt{\log p/n}$ for some $\tau > 2\sqrt{2}$ when $\mathbf{X}$ satisfies a restricted eigenvalue condition.

Bühlmann & Van de Geer (2011) provide further discussion of the restricted eigenvalue condition and other similar conditions on $\mathbf{X}$ to guarantee that the design matrix is well behaved in high dimension.

The theory of lasso suggests that λ is an important factor appearing in its estimation error bound. To achieve optimal estimation error bound, it is desirable to choose the smallest λ such that Equation 4 holds. This choice, however, depends on both the unknown random error distribution and the structure of the design matrix $\mathbf{X}$. As discussed above, a reasonable upper bound for such a theoretical choice of λ requires the knowledge of σ, the standard deviation of the random error. Estimation of σ in high dimension is itself a difficult problem. As a result, it is often infeasible to apply the theoretical choice of λ in real data problems.

2.3. Tuning Parameter Selection via Cross-Validation

In practice, a popular approach to selecting the tuning parameter λ for lasso is a data-driven scheme called cross-validation, which aims for optimal prediction accuracy. Its basic idea is to randomly split the data into a training data set and a testing (or validation) data set such that one may evaluate the prediction error on the testing data while fitting the model using the training data set. There exist several different versions of cross-validation, such as leave-k-out cross-validation, repeated random subsampling validation (also known as Monte Carlo cross-validation), and K-fold cross-validation. Among these options, K-fold cross-validation is most widely applied in real-data analysis.

The steps displayed in Algorithm 1 illustrate the implementation of K-fold cross-validation for lasso. The same idea broadly applies to more general problems such as penalized likelihood estimation with different penalty functions. First, the data are randomly partitioned into K roughly equal-sized subsets (or folds), where the typical choice of K is 5 or 10. Given a value of λ, one of the K folds is retained as the validation data set to evaluate the prediction error, and the remaining data are used as the training data set to obtain $\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda)$. This cross-validation process is then repeated, with each of the K folds used as the validation data set exactly once. For example, in carrying out a 5-fold cross-validation for lasso, we randomly split the data into five roughly equal-sized parts $\mathcal{V}_1, \ldots, \mathcal{V}_5$. Given a tuning parameter λ, we first train the model and estimate $\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda)$ on $\{\mathcal{V}_2, \ldots, \mathcal{V}_5\}$ and then compute the total prediction error on $\mathcal{V}_1$. Repeat this process by training on $\{\mathcal{V}_1, \mathcal{V}_3, \mathcal{V}_4, \mathcal{V}_5\}$ and validating on $\mathcal{V}_2$, and so on. The cross-validation error CV(λ) is obtained as the average of the prediction errors over the K validation data sets from this iterative process.

Algorithm 1 (K-fold cross-validation for lasso).

1. Randomly divide the data of sample size n into K folds, $\mathcal{V}_1, \ldots \mathcal{V}_K$, of roughly equal sizes.
2. Set $\text{Err}(\lambda) = 0$.
3. **for** $k = 1, \ldots, K$.
   4. Training data set $(\mathbf{y}_T, \mathbf{X}_T) = \{(y_i, \mathbf{x}_i) : i \notin \mathcal{V}_k\}$.
   5. Validation data set $(\mathbf{y}_V, \mathbf{X}_V) = \{(y_i, \mathbf{x}_i) : i \in \mathcal{V}_k\}$.
   6. $\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda) \leftarrow \arg\min_{\boldsymbol{\beta}} \{(2|\mathcal{V}_k|)^{-1} ||\mathbf{y}_T - \mathbf{X}_T \boldsymbol{\beta}||^2 + \lambda ||\boldsymbol{\beta}||_1\}$.
   7. $\text{Err}(\lambda) \leftarrow \text{Err}(\lambda) + ||\mathbf{y}_V - \mathbf{X}_V \widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda)||^2$.

   return $\text{CV}(\lambda) = n^{-1}\text{Err}(\lambda)$.

Given a set Λ of candidate tuning parameter values, say, a grid $\{\lambda_1, \ldots, \lambda_m\}$, one would compute CV($\lambda$) according to Algorithm 1 for each $\lambda \in \Lambda$. This yields the cross-validation error curve $\{\text{CV}(\lambda) : \lambda \in \Lambda\}$. To select the optimal λ for lasso, two useful strategies are usually recommended.

A simple and intuitive approach is to select the λ that minimizes the cross-validation error, i.e.,

$$\widehat{\lambda} = \arg\min_{\lambda} \mathrm{CV}(\lambda). \qquad 5.$$

An alternative strategy is based on the so-called one-standard-error rule, which chooses the most parsimonious model (here corresponding to larger λ and more regulation) such that its cross-validation error is within one standard error of $\mathrm{CV}(\widehat{\lambda})$. This is feasible, as the K-fold cross-validation allows one to estimate the standard error of the cross-validation error. The second strategy acknowledges that the cross-validation error curve is estimated with error and is motivated by the principle of parsimony (e.g., section 2.3 of Hastie et al. 2015).

Several R functions are available to implement K-fold cross-validation for lasso, such as the cv.glmnet function in the R package `glmnet` (Friedman et al. 2010) and the cv.lars function in the R package `lars` (Hastie & Efron 2013). Below is the sample R code for performing the 5-fold cross-validation for lasso using the cv.glmnet function.

```
library(glmnet)
data(SparseExample)
cvob1=cv.glmnet(x, y, nfolds=5)
plot(cvob1)
```

The plot produced by the above commands is given in **Figure 1**, which depicts the cross-validation error curve (based on the mean-squared prediction error in this example) as well as the one-standard-error band. In the plot, $\lambda_{\min}$ is the tuning parameter obtained by Equation 5, i.e., the value of the tuning parameter that minimizes the cross-validation prediction error, and $\lambda_{1\mathrm{se}}$

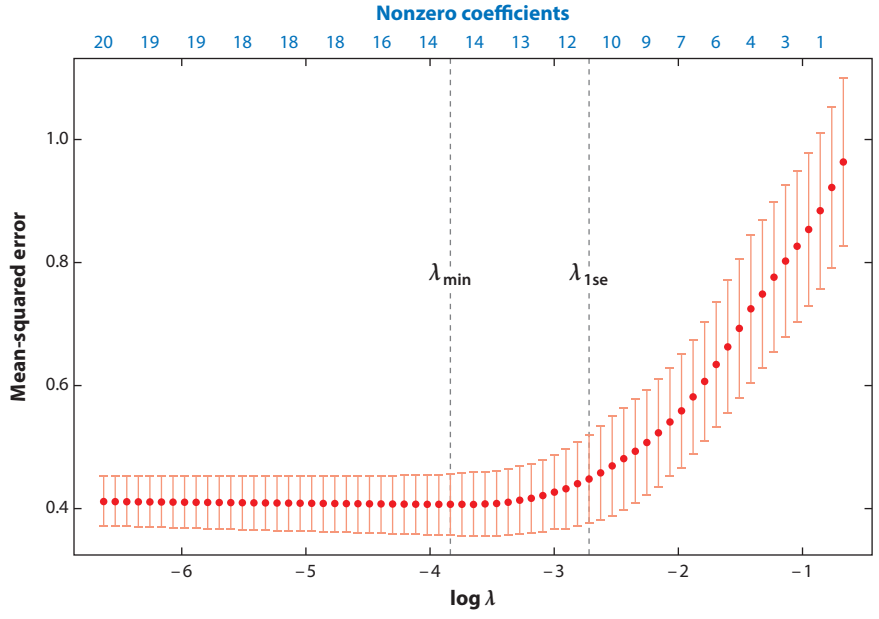

Figure 1

Cross-validation for lasso, including one-standard-error band. $\lambda_{\min}$ is the tuning parameter obtained by Equation 5 that minimizes the cross-validation prediction error, and $\lambda_{1\mathrm{se}}$ is the tuning parameter selected via the one-standard-error rule, and the numbers on the top axis indicate the number of nonzero coefficients for models fitted with different λ values.

denotes the tuning parameter selected via the one-standard-error rule. The numbers at the top of the plot correspond to the numbers of nonzero coefficients (or model sizes) for models fitted with different λ values. For this data example, the prediction errors based on $\lambda_{\min}$ and λ_{1se} are close, while the model based on λ_{1se} is notably sparser than the one based on $\lambda_{\min}$.

In the existing work on the theory of lasso, the tuning parameter is usually considered to be deterministic, or fixed at a prespecified theoretical value. Despite the promising empirical performance of cross-validation, much less is known about the theoretical properties of the cross-validated lasso, where the tuning parameter is selected in a data-driven manner. Some important progress has been made recently in understanding the properties of cross-validated lasso. Homrighausen & McDonald (2013), Chatterjee & Jafarov (2015), and Homrighausen & McDonald (2017) investigated the risk consistency of cross-validated lasso under different regularity conditions. Chetverikov et al. (2016) derived a nonasymptotic error bound for cross-validated lasso and showed that it can achieve the optimal estimation rate up to a factor of order $\sqrt{\log(pn)}$.

2.4. Scaled Lasso: Adapting to an Unknown Noise Level

As discussed earlier, the optimal choice of λ for lasso requires the knowledge of the noise level σ, which is usually unknown in real data analysis. Motivated by Städler et al. (2010) and the discussions on this paper by Antoniadis (2010) and Sun & Zhang (2010), Sun & Zhang (2012) thoroughly investigated the performance of an iterative algorithm called scaled lasso, which jointly estimates the regression coefficients $\boldsymbol{\beta}_0$ and the noise level σ in a sparse linear regression model.

Denote the loss function for lasso regression by

$$L_\lambda(\boldsymbol{\beta}) = (2n)^{-1}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1. \quad 6.$$

The iterative algorithm for scaled lasso is described in Algorithm 2, where $\boldsymbol{\beta}^0$ and λ_0 are initial values independent of $\boldsymbol{\beta}_0$ and σ. In this algorithm, the tuning parameter is rescaled iteratively. In the output of the algorithm, $\widehat{\boldsymbol{\beta}}(\mathbf{X}, \mathbf{y})$ is referred to as the scaled lasso estimator.

Algorithm 2 (Scaled lasso algorithm, Sun & Zhang 2012).

1. Input $(\mathbf{X}, \mathbf{y})$, $\boldsymbol{\beta}^0$, and λ_0.
2. $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}^0$.
3. **while** $L_\lambda(\boldsymbol{\beta}) \le L_\lambda(\boldsymbol{\beta}^0)$
4. $\boldsymbol{\beta}^0 \leftarrow \boldsymbol{\beta}$
5. $\widehat{\sigma} \leftarrow n^{-1/2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^0||$.
6. $\lambda \leftarrow \widehat{\sigma}\lambda_0$.
7. $\boldsymbol{\beta} \leftarrow \arg\min_{\boldsymbol{\beta}} L_\lambda(\boldsymbol{\beta})$.

return $\widehat{\sigma}(\mathbf{X}, \mathbf{y}) \leftarrow \widehat{\sigma}$ and $\widehat{\boldsymbol{\beta}}(\mathbf{X}, \mathbf{y}) \leftarrow \boldsymbol{\beta}^0$.

Sun & Zhang (2012) showed that the outputs of Algorithm 2 converge to the solutions of a joint minimization problem, specifically,

$$(\widehat{\boldsymbol{\beta}}, \widehat{\sigma}) = \arg\min_{\boldsymbol{\beta},\sigma}\left\{(2n\sigma)^{-1}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \sigma/2 + \lambda_0||\boldsymbol{\beta}||_1\right\}. \quad 7.$$

This is equivalent to jointly minimizing Huber's concomitant loss function with the L_1 penalty (Owen 2007, Antoniadis 2010). This loss function possesses the nice property of being jointly convex in $(\boldsymbol{\beta}, \sigma)$. It can also be shown that the solutions are scale equivariant in $\mathbf{y}$, i.e., $\widehat{\boldsymbol{\beta}}(\mathbf{X}, c\mathbf{y}) = c\widehat{\boldsymbol{\beta}}(\mathbf{X}, \mathbf{y})$ and $\widehat{\sigma}(\mathbf{X}, c\mathbf{y}) = |c|\widehat{\sigma}(\mathbf{X}, \mathbf{y})$ for any constant c. This property is practically important in

data analysis. Under the Gaussian assumption and other mild regularity conditions, Sun & Zhang (2012) derived oracle inequalities for prediction and joint estimation of σ and $\boldsymbol{\beta}_0$ for the scaled lasso, which, in particular, imply the consistency and asymptotic normality of $\widehat{\sigma}(\mathbf{X}, \mathbf{y})$ as an estimator for σ.

The function scalreg in the R package `scalreg` implements Algorithm 2 for the scaled lasso. The sample codes below provide an example on how to analyze the sp500 data set in that package with scaled lasso.

```
library(scalreg)
data(sp500)
attach(sp500)
x = sp500.percent[,3: (dim(sp500.percent)[2])]
y = sp500.percent[,1]
scaleob <- scalreg(x, y)
```


3. ALTERNATIVE L_1 PENALTY–BASED METHODS: FROM TUNING SELECTION TO TUNING-FREE

This section provides a brief review of several recently proposed L_1 penalty–based, tuning-free procedures for high-dimensional sparse linear regression. These procedures tackle the challenge of tuning parameter selection for lasso from different angles. As suggested by Equation 4, the theoretically optimal tuning parameter for lasso depends on both the design matrix $\mathbf{X}$ and the unknown error distribution (standard deviation σ, tail behavior, etc.). The three procedures we review here (square-root lasso, TREX, and rank lasso) aim to automatically adapt to one or more aspects of these factors.

3.1. Scale-Free Square-Root Lasso

Square-root lasso is a variant of lasso proposed by Belloni et al. (2011) that enjoys the advantage of avoiding calibrating the tuning parameter with respect to the noise level σ. Square-root lasso replaces least-squares loss (or L_2 loss) function in lasso with its positive square root. Assuming the ϵ_i are independently distributed with mean zero and variance σ^2, the square-root lasso estimator is defined as

$$\widehat{\boldsymbol{\beta}}_{\sqrt{\text{lasso}}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \left\{ n^{-1/2} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}|| + \lambda ||\boldsymbol{\beta}||_1 \right\}. \quad 8.$$

Let $L_{\text{SR}}(\boldsymbol{\beta}) = n^{-1/2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||$ denote the loss function of square-root lasso and let S_{SR} denote its subgradient evaluated at $\boldsymbol{\beta} = \boldsymbol{\beta}_0$. The general principal of tuning parameter selection (e.g., Bickel et al. 2009) suggests choosing λ such that $\text{P}(\lambda > c||S_{\text{SR}}||_\infty) \geq 1 - \alpha_0$ for some constant $c > 1$ and a given small $\alpha_0 > 0$. An important observation that underlies the advantage of square-root lasso is that

$$S_{\text{SR}} = \frac{n^{-1} \sum_{i=1}^n \mathbf{x}_i \epsilon_i}{(n^{-1} \sum_{i=1}^n \epsilon_i^2)^{1/2}}$$

does not depend on σ.

Computationally, the square-root lasso can be formulated as a solution to a convex conic programming problem. The function slim in the R package `flare` (Li et al. 2018) implements a family of lasso variants for high-dimensional regression, including the square-root lasso. The sample

code below demonstrates how to implement the square-root lasso using this function to analyze the sp500 data in the `scalreg` package. The arguments `method=''lq'', q = 2` yield square-root lasso, which are also the default options in the slim function.

```
library(flare)
data(sp500)
attach(sp500)
x = sp500.percent[,3: (dim(sp500.percent)[2])]
y = sp500.percent[,1]
sqrtob <- slim(x, y, method="lq", q = 2)
```

Belloni et al. (2011) recommended the choice $\lambda = cn^{-1/2}\Phi^{-1}\left(1 - \frac{\alpha}{2p}\right)$, for some constant $c > 1$ and $\alpha > 0$. Note that this choice of λ does not depend on σ, and it is valid asymptotically without requiring the random errors to be Gaussian. Under general regularity conditions, Belloni et al. (2011) showed that there exists some positive constant C_n such that

$$\mathrm{P}\left(||\widehat{\boldsymbol{\beta}}_{\sqrt{\text{lasso}}}(\lambda) - \boldsymbol{\beta}_0|| \leq C_n\sigma\left\{n^{-1}s\log(2p/\alpha)\right\}^{1/2}\right) \geq 1 - \alpha,$$

where $s = ||\boldsymbol{\beta}_0||_0$ is the sparsity size of the true model. Hence, square-root lasso achieves the near-oracle rate of lasso even when σ is unknown.

The square-root lasso and lasso are equivalent families of estimators. There exists a one-to-one mapping between the tuning parameter paths of square-root lasso and lasso (Tian et al. 2018). It is also worth pointing out that the square-root lasso is related to but should not be confused with the scaled lasso (Sun & Zhang 2012). The current literature contains some confusion (particularly in the use of names) about these two methods. The connection and distinction between them are nicely discussed by Van de Geer (2016, section 3.7).

3.2. TREX

The scaled lasso and square-root lasso both address the need to calibrate λ for σ. However, the tail behavior of the noise vector, as well as the structure of the design matrix, could also have significant effects on the optimal selection of λ. To alleviate these additional difficulties, Lederer & Müller (2015) proposed a new approach for high-dimensional variable selection. The authors named the new approach TREX to emphasize that it aims at tuning-free (T) regression (R) that adapts to the entire (E) noise and the design matrix $\mathbf{X}$. Indeed, the most attractive property of TREX is that it automatically adjusts λ for the unknown noise standard deviation σ, the tail of the error distribution, and the design matrix.

In contrast to the square-root lasso, the TREX estimator modifies the lasso loss function in a different way. The TREX estimator is defined as

$$\widehat{\boldsymbol{\beta}}_{\text{TREX}} = \arg\min_{\boldsymbol{\beta}}\left\{L_{\text{TREX}}(\boldsymbol{\beta}) + ||\boldsymbol{\beta}||_1\right\}, \tag{9.}$$

where

$$L_{\text{TREX}}(\boldsymbol{\beta}) = \frac{2||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2}{||\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})||_\infty}.$$

TREX does not require a tuning parameter. In this sense, it is a completely tuning-free procedure. Lederer & Müller (2015) proved that the TREX estimator is close to a lasso solution with tuning

parameter of the same order as the theoretically optimal λ. They presented examples where TREX showed promising performance compared with lasso.

The modified loss function for the TREX estimator, however, is no longer convex. Bien et al. (2016) showed the remarkable result that despite the nonconvexity, there exists a polynomial-time algorithm that is guaranteed to find the global minimum of the TREX problem. Bien et al. (2018) recently established a prediction error bound for TREX, which deepens the understanding of the theoretical properties of TREX.

3.3. Rank Lasso: A Tuning-Free and Efficient Procedure

Recently, Wang et al. (2018) proposed an alternative approach to overcoming the challenges of tuning parameter selection for lasso. The new method, named rank lasso, has an optimal tuning parameter that can be easily simulated and automatically adapts to both the unknown random error distribution and the structure of the design matrix. Moreover, it enjoys several other appealing properties: It is a solution to a convex optimization problem and can be conveniently computed via linear programming; it has similar performance to lasso when the random errors are normally distributed and is robust with substantial efficiency gain for heavy-tailed random errors; and it leads to a scale-equivariant estimator that permits coherent interpretation when the response variable undergoes a scale transformation.

Specifically, the new estimator is defined as

$$\widehat{\boldsymbol{\beta}}_{\text{rank}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \left\{ Q_n(\boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_1 \right\}, \tag{10.}$$

where the loss function is

$$Q_n(\boldsymbol{\beta}) = [n(n-1)]^{-1} \sum_{i \neq j} \sum \left| (y_i - \mathbf{x}_i^T \boldsymbol{\beta}) - (y_j - \mathbf{x}_j^T \boldsymbol{\beta}) \right|. \tag{11.}$$

The loss function $Q_n(\boldsymbol{\beta})$ is related to Jaeckel's dispersion function with Wilcoxon scores (Jaeckel 1972) in the classical nonparametric statistics literature. For this reason, the estimator in Equation 10 is referred to as the rank lasso estimator. In the classical low-dimensional setting, regression with Wilcoxon loss function was investigated by Wang (2009) and Wang & Li (2009).

To appreciate its tuning-free property, we observe that the gradient function of $Q_n(\boldsymbol{\beta})$ evaluated at $\boldsymbol{\beta}_0$ is

$$\mathbf{S}_n := \left. \frac{\partial Q_n(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right|_{\boldsymbol{\beta}=\boldsymbol{\beta}_0} = -2[n(n-1)]^{-1} \mathbf{X}^T \boldsymbol{\xi},$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)^T$ with $\xi_i = 2r_i - (n+1)$ and $r_i = \text{rank}(\epsilon_i)$ among $\epsilon_1, \ldots, \epsilon_n$. Note that the random vector $\{r_1, \ldots, r_n\}$ follows the uniform distribution on the permutations of integers $\{1, \ldots, n\}$. Consequently, $\boldsymbol{\xi}$ has a completely known distribution that is independent of the random error distribution. Hence, the gradient function has the complete pivotal property (Parzen et al. 1994), which implies the tuning-free property of rank lasso. To see this, recall that the general principal of tuning parameter selection (Bickel et al. 2009) suggests choosing λ such that $\text{P}(\lambda > c||\mathbf{S}_n||_\infty) \geq 1 - \alpha_0$ for some constant $c > 1$ and a given small $\alpha_0 > 0$. With the design matrix $\mathbf{X}$ and a completely known distribution of $\boldsymbol{\xi}$, we can easily simulate the distribution of $\mathbf{S}_n$ and hence compute the theoretically optimal λ.

Wang et al. (2018) established a finite-sample estimation error bound for the rank lasso estimator with the aforementioned simulated tuning parameter and showed that it achieves the same optimal near-oracle estimation error rate as lasso does. In contrast to lasso, the conditions required by rank lasso for the error distribution are much weaker and allow for heavy-tailed distributions such as Cauchy distribution. Moreover, they proved that further improvement in efficiency can be achieved by a second-stage enhancement with some light tuning.

4. OTHER ALTERNATIVE TUNING PARAMETER SELECTION METHODS FOR LASSO

4.1. Bootstrap-Based Approach

Hall et al. (2009) developed an m-out-of-n bootstrap algorithm to select the tuning parameter for lasso, pointing out that standard bootstrap methods would fail. Their algorithm employs a wild bootstrap procedure (see Algorithm 3), which allows one to estimate the mean-squared error of the parameter estimators for different tuning parameters. For each candidate λ, this algorithm computes the bootstrapped mean-squared error estimate $\mathrm{Err}(\lambda)$. The optimal tuning parameter is chosen as $\widehat{\lambda}_{\mathrm{boots}} = (n/m)^{1/2} \arg\min_{\lambda} \mathrm{Err}(\lambda)$. The final estimator for $\boldsymbol{\beta}_0$ is given by

$$\widehat{\boldsymbol{\beta}}_{\mathrm{boots}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \bar{y} - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \widehat{\lambda}_{\mathrm{boots}} ||\boldsymbol{\beta}||_1 \right\}.$$

Algorithm 3 (Bootstrap algorithm).

1. Input $(\mathbf{y}, \mathbf{X})$, a $\sqrt{n}$-consistent "pilot estimator" $\widetilde{\boldsymbol{\beta}}$, and λ.
2. $\widehat{\epsilon}_i \leftarrow y_i - \bar{y} - \mathbf{x}_i^T \widetilde{\boldsymbol{\beta}}$.
3. $\widetilde{\epsilon}_i \leftarrow \widehat{\epsilon}_i - n^{-1} \sum_{j=1}^{n} \widehat{\epsilon}_j$.
4. Set $\mathrm{Err}(\boldsymbol{\lambda}) \leftarrow 0$.
5. **For** $\mathrm{k} = 1, \ldots, N$
   6. Obtain $\epsilon_1^*, \ldots, \epsilon_m^*$ by sampling randomly from $\widetilde{\epsilon}_1, \ldots, \widetilde{\epsilon}_n$ with replacement.
   7. $y_i^* \leftarrow \bar{y} + \mathbf{x}_i^T \widetilde{\boldsymbol{\beta}} + \epsilon_i^*, i = 1, \ldots, m$.
   8. $\widehat{\boldsymbol{\beta}}^*(\lambda) \leftarrow \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{m} (y_i^* - \bar{y}^* - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda ||\boldsymbol{\beta}||_1 \right\}$.
   9. $\mathrm{Err}(\lambda) \leftarrow \mathrm{Err}(\lambda) + ||\widehat{\boldsymbol{\beta}}^*(\lambda) - \widetilde{\boldsymbol{\beta}}||^2$.

   return $\mathrm{Err}(\lambda)$.

Their method and theory were mostly developed for the $p < n$ case. The algorithm requires that the covariates are centered at their empirical means and that a $\sqrt{n}$-consistent pilot estimator $\widetilde{\boldsymbol{\beta}}$ is available. Hall et al. (2009) proved that if $m = O(n/(\log n)^{1+\eta})$ for some $\eta > 0$, then the estimator $\widehat{\boldsymbol{\beta}}_{\mathrm{boots}}$ can identify the true model with probability approaching one as $n \to \infty$. They also suggested that the theory can be generalized to the high-dimensional case with fixed sparsity $||\boldsymbol{\beta}_0||_0$; however, the order of p would depend on the generalized parameters of the model such as the tail behaviors of the random noise.

Chatterjee & Lahiri (2011) proposed a modified bootstrap method for lasso. This method first computes a thresholded version of the lasso estimator and then applies the residual bootstrap. In the classical $p \ll n$ setting, Chatterjee & Lahiri (2011) proved that the modified bootstrap method provides a valid approximation of the distribution of the lasso estimator. They further recommended choosing λ to minimize the bootstrapped approximation to the mean-squared error of the lasso estimator.

4.2. Adaptive Calibration for l_∞

Motivated by Lepski's method for nonparametric regression (Lepski 1991, Lepski & Spokoiny 1997), Chichignoud et al. (2016) proposed a novel adaptive validation method for tuning parameter selection for lasso. The method, called adaptive calibration for l_∞, or AV_∞, performs simple tests along a single lasso path to select the optimal tuning parameter. The method is equipped with a fast computational routine and theoretical guarantees on its finite-sample performance with respect to the supernorm loss.

Let $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$ be a set of candidate values for λ, where $0 < \lambda_1 < \cdots < \lambda_N = \lambda_{\max} = 2n^{-1}||\mathbf{X}^T\mathbf{y}||_\infty$. Denote $\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda_j)$ as the lasso estimator in Equation 3, with the tuning parameter set as $\lambda = \lambda_j$, $j = 1, \ldots, N$. The proposed AV_∞ selects λ based on the tests for supremum norm differences of lasso estimates with different tuning parameters. It is defined as

$$\widehat{\lambda}_{\mathrm{AC}} = \min\left\{\lambda \in \Lambda : \max_{\substack{\lambda', \lambda'' \in \Lambda \\ \lambda', \lambda'' \geq \lambda}} \left[\frac{||\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda') - \widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda'')||_\infty}{\lambda' + \lambda''} - \bar{C}\right] \leq 0\right\}, \tag{12.}$$

where $\bar{C}$ is a constant with respect to the L_∞ error bound of lasso estimator. Chichignoud et al. (2016) recommended the universal choice $\bar{C} = 0.75$ for all practical purposes.

Chichignoud et al. (2016) proposed a simple and fast implementation for the tuning parameter selection via AV_∞; see the description in Algorithm 4, where in the algorithm the binary random variable $\widehat{t}_{\lambda_j}$ is defined as

$$\widehat{t}_{\lambda_j} = \prod_{k=j}^{N} \mathbf{1}\left\{\frac{||\widehat{\boldsymbol{\beta}}(\lambda_j) - \widehat{\boldsymbol{\beta}}(\lambda_k)||_\infty}{\lambda_j + \lambda_k} \leq \bar{C}\right\}, \quad j = 1, \ldots, N,$$

with $\mathbf{1}$ being the indicator function. The final estimator for the AV_∞ method is the lasso estimator with the tuning parameter $\widehat{\lambda}_{\mathrm{AC}}$, denoted as $\widehat{\boldsymbol{\beta}}(\widehat{\lambda}_{\mathrm{AC}})$. As shown in Algorithm 4, AV_∞ only needs to compute one solution path, in contrast to the K paths in the K-fold cross-validation for lasso. The new method is usually faster than cross-validation. Chichignoud et al. (2016) proved that $||\widehat{\boldsymbol{\beta}}(\widehat{\lambda}_{\mathrm{AC}}) - \boldsymbol{\beta}_0||_\infty$ achieves the optimal supremum norm error bound of lasso up to a constant prefactor with high probability under some regularity conditions.

Algorithm 4 (AV_∞ algorithm).

1. Input $\widehat{\boldsymbol{\beta}}(\lambda_1), \ldots, \widehat{\boldsymbol{\beta}}(\lambda_N), \bar{C}$.
2. Set $j \leftarrow N$.
3. **While** $\widehat{t}_{\lambda_{j-1}} \neq 0$ and $j > 1$
4. Update index $j \leftarrow j - 1$.

   return $\widehat{\lambda} \leftarrow \lambda_j$.

5. NONCONVEX PENALIZED HIGH-DIMENSIONAL REGRESSION AND TUNING FOR SUPPORT RECOVERY

5.1. Background

Lasso is known to achieve accurate prediction under rather weak conditions (Greenshtein & Ritov 2004). However, it is also widely recognized that lasso requires stringent conditions on the design matrix $\mathbf{X}$ to achieve variable selection consistency (Zhao & Yu 2006, Zou 2006). In many scientific problems, it is of importance to identify relevant or active variables. For example, biologists are

often interested in identifying the genes associated with certain disease. This problem is often referred to as support recovery, with the goal of identifying $\mathcal{S}_0 = \{j : \beta_{0j} \neq 0, j = 1, \ldots, p\}$.

To alleviate the bias of lasso due to the overpenalization of L_1 penalty, nonconvex penalized regression has been studied in the literature as an alternative to lasso (Fan & Lv 2010, Zhang & Zhang 2012). Two popular choices of nonconvex penalty functions are SCAD (Fan & Li 2001) and MCP (C.H. Zhang 2010). The SCAD penalty function is given by

$$p_\lambda(|\beta_j|) = \begin{cases} \lambda|\beta_j|, & \text{if } |\beta_j| \le \lambda, \\ \frac{2a\lambda|\beta_j| - \beta_j^2 - \lambda^2}{2(a-1)}, & \text{if } \lambda < |\beta_j| < a\lambda, \\ \frac{(a+1)\lambda^2}{2}, & \text{if } |\beta_j| \ge a\lambda, \end{cases} \quad 13.$$

where $a > 2$ is a constant and Fan & Li (2001) recommended the choice $a = 3.7$. The MCP penalty function is given by

$$p_\lambda(|\beta_j|) = \begin{cases} \lambda|\beta_j| - \frac{\beta_j^2}{2a}, & \text{if } |\beta_j| \le a\lambda, \\ \frac{a\lambda^2}{2}, & \text{if } |\beta_j| > a\lambda. \end{cases} \quad 14.$$

where $a > 1$ is a constant. **Figure 2** depicts the two penalty functions.

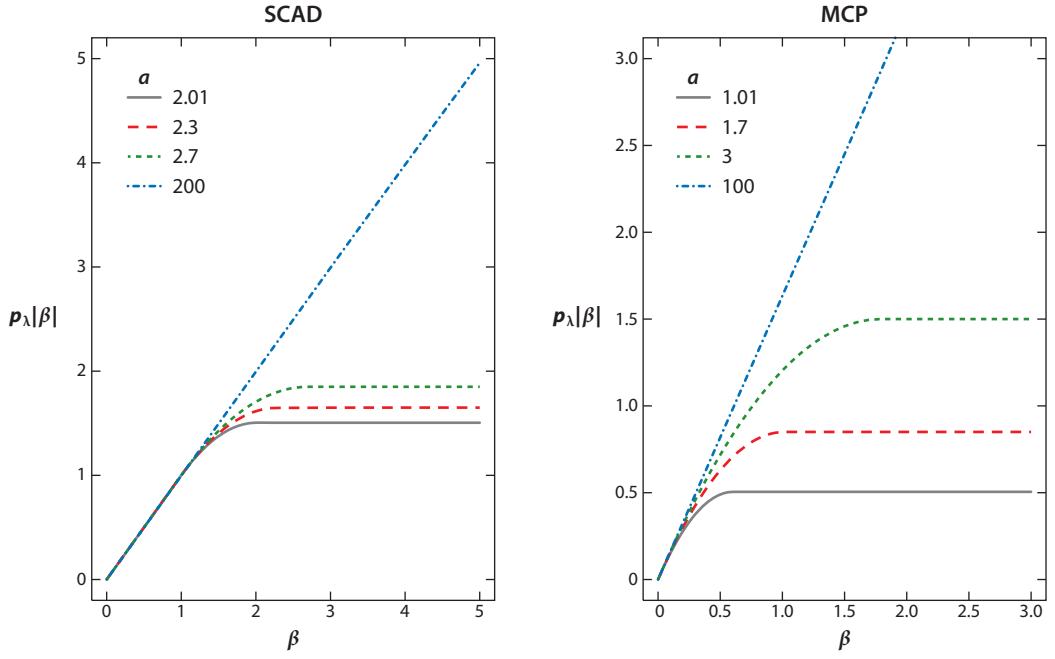

Figure 2

SCAD and MCP penalty functions ($\lambda = 1$). a is a constant, p_λ is the penalty function, and $\boldsymbol{\beta}$ is an estimator. Abbreviations: MCP, minimax concave penalty; SCAD, smoothly clipped absolute deviation.

As cross-validation for lasso aims for prediction accuracy, it tends to select a somewhat smaller tuning parameter (i.e., less regulation). The resulting model size hence is usually larger than the true model size. In the fixed p setting, Wang et al. (2007) proved that with a positive probability cross-validation leads to a tuning parameter that would yield an overfitted model.

Recent research has shown that when nonconvex penalized regression is combined with some modified Bayesian information criterion– (BIC-) type criterion, the underlying model can be identified with probability approaching one under appropriate regularity conditions. Several useful results were obtained in the low-dimensional setting. For example, effective BIC-type criteria for tuning parameter selection for nonconvex penalized regression were investigated in Wang et al. (2007) for fixed p and Wang et al. (2009) for diverging p (but $p < n$). Zou et al. (2007) considered Akaike information criterion– (AIC-) and BIC-type criteria based on the degrees of freedom for lasso. Also in the fixed p setting, Zhang et al. (2010) studied the generalized information criterion, encompassing AIC and BIC. They revealed that a BIC-type selector enables identification of the true model consistently and that an AIC-type selector is asymptotically loss efficient.

In the rest of this section, we review several modified BIC-type criteria in the high-dimensional setup ($p \gg n$) for tuning parameter selection with the goal of support recovery.

5.2. Extended Bayesian Information Criterion for Comparing Models When $p \gg n$

Let $\mathcal{S}$ be an arbitrary subset of $\{1, \ldots, p\}$. Hence, each $\mathcal{S}$ indexes a candidate model. Given the data $(\mathbf{X}, \mathbf{y})$, the classical BIC, proposed by Schwarz (1978), is defined as follows:

$$\mathrm{BIC}(\mathcal{S}) = -2 \log L_n\{\widehat{\boldsymbol{\beta}}(\mathcal{S})\} + ||\mathcal{S}||_0 \log n,$$

where $L_n(\cdot)$ is the likelihood function, $\widehat{\boldsymbol{\beta}}(\mathcal{S})$ is the maximum likelihood estimator for the model with support $\mathcal{S}$, and $||\mathcal{S}||_0$ is the cardinality of the set $\mathcal{S}$. Given different candidate models, BIC selects the model with support $\mathcal{S}$ such that $\mathrm{BIC}(\mathcal{S})$ is minimized.

In the classical framework where p is small and fixed, it is known (Rao & Wu 1989) that under standard conditions BIC is variable selection consistent, i.e., $\mathcal{S}_0$ is identified with probability approaching one as $n \to \infty$ if the true model is in the set of candidate models. However, in the large p setting, the number of candidate models grows exponentially fast in p. The classical BIC is no longer computationally feasible.

Chen & Chen (2008) were the first to rigorously study the extension of BIC to high-dimensional regression where $p \gg n$. They proposed an extended family of BIC of the form

$$\mathrm{BIC}_\gamma(\mathcal{S}) = -2 \log L_n\{\widehat{\boldsymbol{\beta}}(\mathcal{S})\} + ||\mathcal{S}||_0 \log n + 2\gamma \log \binom{p}{||\mathcal{S}||_0}, \qquad 15.$$

where $\gamma \in [0, 1]$. Comparing with the classical BIC, the above modification incorporates the model size in the penalty term. It was proved that if $p = O(n^\kappa)$ for some constant κ, and $\gamma > 1 - (2\kappa)^{-1}$, then this extended BIC is variable selection consistent under some regularity conditions. Kim et al. (2012) also investigated variants of extended BIC for comparing models for high-dimensional least-squares regression.

5.3. High-Dimensional Bayesian Information Criterion for Tuning Parameter Selection and Support Recovery

The extended BIC is most useful if a candidate set of models is provided and if the true model is contained in such a candidate set with high probability. One practical choice is to construct

such a set of candidate models from a lasso solution path. As lasso requires stringent conditions on the design matrix $\mathbf{X}$ to be variable selection consistent, it is usually not guaranteed that the lasso solution path contains the oracle estimator, the estimator corresponding to support set $\mathcal{S}_0$. Alternatively, one may construct a set of candidate models from the solution path of SCAD or MCP. However, as the objective function of SCAD or MCP is nonconvex, multiple minima may be present. The solution path of SCAD or MCP, hence, may be nonunique and does not necessarily contain the oracle estimator. Even if a solution path is known to contain the oracle estimator, finding the optimal tuning parameter that yields the oracle estimator with theoretical guarantee is challenging in high dimension.

To overcome these difficulties, Wang et al. (2013) thoroughly studied how to calibrate nonconvex penalized least-squares regression to find the optimal tuning parameter for support recovery when $p \gg n$. Define a consistent solution path to be a path that contains the oracle estimator with probability approaching one. Wang et al. (2013) first proved that an easy-to-calculate calibrated CCCP (which stands for ConCave Convex Procedure) algorithm produces a consistent solution path. Furthermore, they proposed HBIC, a high-dimensional BIC criterion, and proved that it can be applied to the solution path to select the optimal tuning parameter that asymptotically identifies the oracle estimator. Let $\widetilde{\boldsymbol{\beta}}(\lambda)$ be the solution corresponding to λ on a consistent solution path, for example, the one obtained by the aforementioned calibrated nonconvex penalized regression with SCAD or MCP penalty. HBIC selects the optimal tuning parameter λ in $\Lambda_n = \{\lambda : ||\widetilde{\boldsymbol{\beta}}(\lambda)||_0 \leq K_n\}$, where K_n is allowed to diverge to infinity, by minimizing

$$\mathrm{HBIC}(\lambda) = \log\left\{\frac{1}{n}||\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}(\lambda)||^2\right\} + ||\widetilde{\boldsymbol{\beta}}(\lambda)||_0 \frac{C_n \log p}{n}, \qquad 16.$$

where C_n diverges to infinity. Wang et al. (2013) proved that if $C_n||\boldsymbol{\beta}_0||_0 \log p = o(n)$ and $K_n^2 \log p \log n = o(n)$, then under mild conditions, HBIC identifies the true model with probability approaching one. For example, one can take $C_n = \log(\log n)$. Note that the consistency is valid in the ultrahigh-dimensional setting, where p is allowed to grow exponentially fast in n.

In addition, Wang & Zhu (2011) studied a variant of HBIC in combination with a sure screening procedure. Fan & Tang (2013) investigated the proxy generalized information criterion (a proxy of the generalized information criterion) (Zhang et al. 2010), when $p \gg n$. They identified a range of complexity penalty levels such that the tuning parameter that is selected by optimizing the proxy generalized information criterion can achieve model selection consistency.

6. A REAL DATA EXAMPLE

We consider the data set sp500 in the R package `scalreg`, which contains a year's worth of close-of-day data for most of the Standard & Poor's S&P 500 stocks. The response variable `sp500.percent` is the daily percentage change. The data set has 252 observations of 492 variables.

We demonstrate the performance of lasso with K-fold cross-validation, scaled lasso, and square-root lasso methods on this example. Other methods reviewed in this article that do not yet have publicly available software packages are not implemented. We evaluate the performance of different methods based on 100 random splits. For each split, we randomly select half of the data to train the model, and then compute the L_1- and L_2-prediction errors and estimated model sizes on the other half of the data. For lasso, we select the tuning parameter by 10-fold cross-validation, using the R function cv.glmnet and the one-standard-error rule. For scaled lasso, we apply the default tuning parameter selection method in the R function scalreg, which is the quantile-based penalty level (lam0="quantile") introduced and studied in Sun & Zhang (2013). For the

Table 1 Analysis of S&P 500 data set

	Lasso	Scaled lasso	Square-root lasso
L_1 error	0.17 (0.02)	0.21 (0.02)	0.17 (0.02)
L_2 error	0.05 (0.01)	0.08 (0.03)	0.05 (0.01)
Sparsity	60.03 (5.39)	120.82 (4.70)	76.63 (8.27)

Quantities in parentheses represent the standard deviations. Abbreviation: S&P, Standard & Poor's.

square-root lasso method, we use R function slim to train the model. However, the package does not have a built-in tuning parameter selection method. As the optimal tuning parameter depends on the tail behavior of the random error, it is also chosen by 10-fold cross-validation.

Table 1 summarizes the averages and standard deviations of the L_1- and L_2-prediction errors and estimated model sizes for the three methods with 100 random splits. Lasso and square-root lasso have similar performance, though lasso method tends to yield sparser models. Scaled lasso has slightly larger prediction errors and model sizes. The difference may be due to the nonnormality of the data, which would affect the performance of the default tuning parameter selection method in the scalreg function.

7. DISCUSSION

Developing computationally efficient tuning parameter selection methods with theoretical guarantees is important for many high-dimensional statistical problems but has so far only received limited attention in the current literature. This article reviews several commonly used tuning parameter selection approaches for high-dimensional linear regression and provides some insights on how they work. The aim is to bring more attention to this important topic to help stimulate future fruitful research in this direction.

This review article focuses on regularized least-squares types of estimation procedures for sparse linear regression. The specific choice of tuning parameter necessarily depends on the user's own research objectives: Is prediction the main research goal? Or is identifying relevant variables of more importance? How much computational time is the researcher willing to allocate? Is robustness of any concern for the data set under consideration?

The problem of tuning parameter selection is ubiquitous and has been investigated in settings beyond sparse linear least-squares regression. Lee et al. (2014) extended the idea of extended BIC to high-dimensional quantile regression. They recommended selecting the model that minimizes

$$\mathrm{BIC}_{\mathrm{Q}}(\mathcal{S}) = \log\left\{\sum_{i=1}^{n}\rho_\tau\left(y_i - \mathbf{x}_i^T\widehat{\boldsymbol{\beta}}(\mathcal{S})\right)\right\} + (2n)^{-1}C_n||\mathcal{S}||_0\log n, \qquad 17.$$

where $\rho_\tau(u) = 2u(\tau - I(u<0))$ is the quantile loss function, and C_n is some positive constant that diverges to infinity as n increases. They also proved the variable selection consistency property when $C_n \log n/n \to 0$ under some regularity conditions. Belloni & Chernozhukov (2011) and Koenker (2011) considered tuning parameter selection for penalized quantile regression based on the pivotal property of the quantile score function. Wang et al. (2012) considered tuning parameter selection using cross-validation with the quantile loss function. For support vector machines, a widely used approach for classification, Zhang et al. (2016) recently established the consistency of extended BIC-type criterion for tuning parameter selection in the high-dimensional setting. For semiparametric regression models, Xie & Huang (2009) explored cross-validation for

high-dimensional partially linear mean regression; Sherwood & Wang (2016) applied an extended BIC-type criterion for high-dimensional partially linear additive quantile regression. Datta & Zhou (2017) derived a corrected cross-validation procedure for high-dimensional linear regression with error in variables. Guo et al. (2016) used an extended BIC-type criterion for high-dimensional and banded vector autoregressions. In studying high-dimensional panel data, Kock (2013) empirically investigated both cross-validation and BIC for tuning parameter selection.

Although the basic ideas of cross-validation and BIC can be intuitively generalized to more complex modeling settings, their theoretical justifications are often still lacking despite the promising numerical evidence. It is worth emphasizing that intuition is not always straightforward and theoretical insights can be valuable. For instance, when investigating high-dimensional graphs and variable selection with lasso, Meinshausen & Bühlmann (2006) observed that the consistency of neighborhood selection hinges on the choice of the penalty parameter. The oracle value for optimal prediction does not lead to a consistent neighborhood estimate.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

The authors thank the editors for their interest in this topic and an anonymous referee for constructive comments. They acknowledge the support of VA IIR 16-253 and NSF DMS-1712706.

LITERATURE CITED

Antoniadis A. 2010. Comments on: l_1-penalization for mixture regression models. *TEST* 19:257–58

Belloni A, Chernozhukov V. 2011. l_1-penalized quantile regression in high-dimensional sparse models. *Ann. Stat.* 39:82–130

Belloni A, Chernozhukov V, Wang L. 2011. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika* 98:791–806

Bickel PJ, Ritov Y, Tsybakov AB. 2009. Simultaneous analysis of Lasso and Dantzig selector. *Ann. Stat.* 37:1705–32

Bien J, Gaynanova I, Lederer J, Müller C. 2016. Non-convex global minimization and false discovery rate control for the TREX. arXiv:1604.06815 [stat.ML]

Bien J, Gaynanova I, Lederer J, Müller CL. 2018. Prediction error bounds for linear regression with the TREX. *TEST* :1–24

Boyd S, Vandenberghe L. 2004. *Convex Optimization*. Cambridge, UK: Cambridge Univ. Press

Bühlmann P, Van de Geer S. 2011. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. New York: Springer

Bunea F, Tsybakov A, Wegkamp M. 2007. Sparsity oracle inequalities for the Lasso. *Electron. J. Stat.* 1:169–94

Candes E, Tao T. 2007. The Dantzig selector: statistical estimation when p is much larger than n. *Ann. Stat.* 35:2313–51

Candès EJ, Plan Y. 2009. Near-ideal model selection by l_1 minimization. *Ann. Stat.* 37:2145–77

Chatterjee A, Lahiri SN. 2011. Bootstrapping Lasso estimators. *J. Am. Stat. Assoc.* 106:608–25

Chatterjee S, Jafarov J. 2015. Prediction error of cross-validated Lasso. arXiv:1502.06291 [math.ST]

Chen J, Chen Z. 2008. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika* 95:759–71

Chen SS, Donoho DL, Saunders MA. 2001. Atomic decomposition by basis pursuit. *SIAM Rev.* 43:129–59

Chetverikov D, Liao Z, Chernozhukov V. 2016. On cross-validated Lasso. arXiv:1605.02214 [math.ST]

Chichignoud M, Lederer J, Wainwright M. 2016. A practical scheme and fast algorithm to tune the lasso with optimality guarantees. *J. Mach. Learn. Res.* 17:1–20
Datta A, Zou H. 2017. CoCoLasso for high-dimensional error-in-variables regression. *Ann. Stat.* 45:2400–26
Efron B, Hastie T, Johnstone I, Tibshirani R. 2004. Least angle regression. *Ann. Stat.* 32:407–99
Fan J, Li R. 2001. Variable selection via nonconcave penalized likelihood and its oracle property. *J. Am. Stat. Assoc.* 96:1348–60
Fan J, Lv J. 2010. A selective overview of variable selection in high dimensional feature space. *Stat. Sin.* 20:101–48
Fan Y, Tang CY. 2013. Tuning parameter selection in high dimensional penalized likelihood. *J. R. Stat. Soc. B* 75:531–52
Friedman J, Hastie T, Höfling H, Tibshirani R. 2007. Pathwise coordinate optimization. *Ann. Appl. Stat.* 1:302–32
Friedman J, Hastie T, Tibshirani R. 2010. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* 33:1–22
Greenshtein E, Ritov Y. 2004. Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli* 10:971–88
Guo S, Wang Y, Yao Q. 2016. High-dimensional and banded vector autoregressions. *Biometrika* 103:889–903
Hall P, Lee ER, Park BU. 2009. Bootstrap-based penalty choice for the lasso, achieving oracle performance. *Stat. Sin.* 19:449–71
Hastie T, Efron B. 2013. `lars`: least angle regression, lasso and forward stagewise. *R package*, version 1.2. **https://cran.r-project.org/web/packages/lars/index.html**
Hastie T, Tibshirani R, Friedman J. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer. 2nd ed.
Hastie T, Tibshirani R, Wainwright M. 2015. *Statistical Learning with Sparsity: the Lasso and Generalizations*. Boca Raton, FL: Chapman and Hall/CRC
Homrighausen D, McDonald DJ. 2013. The lasso, persistence, and cross-validation. In *Proceedings of the 30th International Conference on Machine Learning*, ed. S Dasgupta, D McAllester, pp. 1031–39. New York: ACM
Homrighausen D, McDonald DJ. 2017. Risk consistency of cross-validation with lasso-type procedures. *Stat. Sin.* 27:1017–36
Jaeckel LA. 1972. Estimating regression coefficients by minimizing the dispersion of the residuals. *Ann. Math. Stat.* 43:1449–58
Kim Y, Kwon S, Choi H. 2012. Consistent model selection criteria on high dimensions. *J. Mach. Learn. Res.* 13:1037–57
Kock AB. 2013. Oracle efficient variable selection in random and fixed effects panel data models. *Econom. Theory* 29:115–52
Koenker R. 2011. Additive models for quantile regression: model selection and confidence bandaids. *Braz. J. Probab. Stat.* 25:239–62
Lederer J, Müller C. 2015. Don't fall for tuning parameters: tuning-free variable selection in high dimensions with the TREX. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2729–35. Palo Alto, CA: AAAI
Lee ER, Noh H, Park BU. 2014. Model selection via Bayesian information criterion for quantile regression models. *J. Am. Stat. Assoc.* 109:216–29
Lepski OV. 1991. On a problem of adaptive estimation in Gaussian white noise. *Theory Probab. Appl.* 35:454–66
Lepski OV, Spokoiny VG. 1997. Optimal pointwise adaptive methods in nonparametric estimation. *Ann. Stat.* 25:2512–46
Li X, Zhao T, Wang L, Yuan X, Liu H. 2018. `flare`: family of lasso regression. *R package*, version 1.6.0. **https://cran.r-project.org/web/packages/flare/index.html**
Meinshausen N, Bühlmann P. 2006. High-dimensional graphs and variable selection with the Lasso. *Ann. Stat.* 34:1436–62
Negahban SN, Ravikumar P, Wainwright MJ, Yu B. 2012. A unified framework for high-dimensional analysis of *m*-estimators with decomposable regularizers. *Stat. Sci.* 27:538–57

Owen AB. 2007. A robust hybrid of lasso and ridge regression. *Contemp. Math* 443:59–71
Parzen M, Wei L, Ying Z. 1994. A resampling method based on pivotal estimating functions. *Biometrika* 81:341–50
Rao R, Wu Y. 1989. A strongly consistent procedure for model selection in a regression problem. *Biometrika* 76:369–74
Schwarz G. 1978. Estimating the dimension of a model. *Ann. Stat.* 6:461–64
Sherwood B, Wang L. 2016. Partially linear additive quantile regression in ultra-high dimension. *Ann. Stat.* 44:288–317
Städler N, Bühlmann P, Van De Geer S. 2010. l_1-penalization for mixture regression models. *TEST* 19:209–56
Sun T, Zhang CH. 2010. Comments on: l_1-penalization for mixture regression models. *TEST* 19:270–75
Sun T, Zhang CH. 2012. Scaled sparse linear regression. *Biometrika* 99:879–98
Sun T, Zhang CH. 2013. Sparse matrix inversion with scaled Lasso. *J. Mach. Learn. Res.* 14:3385–418
Tian X, Loftus JR, Taylor JE. 2018. Selective inference with unknown variance via the square-root lasso. *Biometrika* 105:755–68
Tibshirani R. 1996. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B* 58:267–88
Van de Geer S. 2016. *Estimation and Testing Under Sparsity*. New York: Springer
Van de Geer SA. 2008. High-dimensional generalized linear models and the lasso. *Ann. Stat.* 36:614–45
Wainwright M. 2019. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint.* Cambridge, UK: Cambridge Univ. Press
Wang H, Li B, Leng C. 2009. Shrinkage tuning parameter selection with a diverging number of parameters. *J. R. Stat. Soc. B* 71:671–83
Wang H, Li R, Tsai CL. 2007. Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika* 94:553–68
Wang L. 2009. Wilcoxon-type generalized Bayesian information criterion. *Biometrika* 96:163–73
Wang L, Kim Y, Li R. 2013. Calibrating nonconvex penalized regression in ultra-high dimension. *Ann. Stat.* 41:2505–36
Wang L, Li R. 2009. Weighted Wilcoxon-type smoothly clipped absolute deviation method. *Biometrics* 65:564–71
Wang L, Peng B, Bradic J, Li R, Wu Y. 2018. *A tuning-free robust and efficient approach to high-dimensional regression*. Tech. Rep., Sch. Stat., Univ. Minn.
Wang L, Wu Y, Li R. 2012. Quantile regression for analyzing heterogeneity in ultra-high dimension. *J. Am. Stat. Assoc.* 107:214–22
Wang T, Zhu L. 2011. Consistent tuning parameter selection in high dimensional sparse linear regression. *J. Multivar. Anal.* 102:1141–51
Wu TT, Lange K. 2008. Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.* 2:224–44
Xie H, Huang J. 2009. SCAD-penalized regression in high-dimensional partially linear models. *Ann. Stat.* 37:673–96
Zhang CH. 2010. Nearly unbiased variable selection under minimax concave penalty. *Ann. Stat.* 38:894–942
Zhang CH, Huang J. 2008. The sparsity and bias of the lasso selection in high-dimensional linear regression. *Ann. Stat.* 36:1567–94
Zhang CH, Zhang T. 2012. A general theory of concave regularization for high-dimensional sparse estimation problems. *Stat. Sci.* 27:576–93
Zhang T. 2010. Analysis of multi-stage convex relaxation for sparse regularization. *J. Mach. Learn. Res.* 11:1081–107
Zhang X, Wu Y, Wang L, Li R. 2016. A consistent information criterion for support vector machines in diverging model spaces. *J. Mach. Learn. Res.* 17:466–91
Zhang Y, Li R, Tsai CL. 2010. Regularization parameter selections via generalized information criterion. *J. Am. Stat. Assoc.* 105:312–23
Zhao P, Yu B. 2006. On model selection consistency of lasso. *J. Mach. Learn. Res.* 7:2541–63
Zou H. 2006. The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.* 101:1418–29
Zou H, Hastie T, Tibshirani R. 2007. On the "degrees of freedom" of the lasso. *Ann. Stat.* 35:2173–92

Annual Review of Statistics and Its Application

Volume 7, 2020

Contents

Errata

An online log of corrections to *Annual Review of Statistics and Its Application* articles may be found at http://www.annualreviews.org/errata/statistics